IN THE UNITED STATES DISTRICT COURT
FOR THE EASTERN DISTRICT OF TEXAS
MARSHALL DIVISION

MICROUNITY SYSTEMS ENGINEERING, INC.	§	
	§	
v.	§	Case No. 2:10-CV-91-LED-RSP
	§	
ACER, INC., et al.	§	

CLAIM CONSTRUCTION
MEMORANDUM AND ORDER

On August 16, 2012, the Court held a claim construction hearing concerning a family of related patents: U.S. Patent Nos. 5,742,840; 5,794,061; 6,006,318; 6,643,765; 6,725,356; 7,213,131; 7,216,217; 7,353,367; 7,509,366; 7,653,806; 7,660,972; 7,660,973; and 7,730,287. The Court issued a provisional claim construction order on August 18, 2012. (Dkt. No. 480.) Having considered the arguments and evidence presented by the parties at the hearing and in their briefing (Dkt. Nos. 443, 467, and 474), the Court issues this Claim Construction Order.

APPLICABLE LAW

"It is a 'bedrock principle' of patent law that 'the claims of a patent define the invention to which the patentee is entitled the right to exclude.'" *Phillips v. AWH Corp.*, 415 F.3d 1303, 1312 (Fed. Cir. 2005) (en banc) (quoting *Innova/Pure Water Inc. v. Safari Water Filtration Sys., Inc.*, 381 F.3d 1111, 1115 (Fed. Cir. 2004)). To determine the meaning of the claims, courts start by considering the intrinsic evidence. *See id.* at 1313. *C.R. Bard, Inc. v. U.S. Surgical Corp.*, 388 F.3d 858, 861 (Fed. Cir. 2004); *Bell Atl. Network Servs., Inc. v. Covad Commc'ns Group, Inc.*, 262 F.3d 1258, 1267 (Fed. Cir. 2001). The intrinsic evidence includes the claims themselves, the specification, and the prosecution history. *See Phillips*, 415 F.3d at 1314; *C.R. Bard, Inc.*, 388 F.3d at 861. Courts give claim terms their ordinary and accustomed meaning as understood by one of ordinary skill in the art at the time of the invention in the context of the

entire patent. *Phillips*, 415 F.3d at 1312–13; *Alloc, Inc. v. Int'l Trade Comm'n*, 342 F.3d 1361, 1368 (Fed. Cir. 2003).

The claims themselves provide substantial guidance in determining the meaning of particular claim terms. *Phillips*, 415 F.3d at 1314. First, a term's context in the asserted claim can be very instructive. *Id*. Other asserted or unasserted claims can also aid in determining the claim's meaning because claim terms are typically used consistently throughout the patent. *Id*. Differences among the claim terms can also assist in understanding a term's meaning. *Id*. For example, when a dependent claim adds a limitation to an independent claim, it is presumed that the independent claim does not include the limitation. *Id*. at 1314–15.

"[C]laims 'must be read in view of the specification, of which they are a part.'" *Id*. (quoting *Markman v. Westview Instruments, Inc.*, 52 F.3d 967, 979 (Fed. Cir. 1995) (en banc)). "[T]he specification 'is always highly relevant to the claim construction analysis. Usually, it is dispositive; it is the single best guide to the meaning of a disputed term.'" *Id*. (quoting *Vitronics Corp. v. Conceptronic, Inc.*, 90 F.3d 1576, 1582 (Fed. Cir. 1996)); *Teleflex, Inc. v. Ficosa N. Am. Corp.*, 299 F.3d 1313, 1325 (Fed. Cir. 2002). This is true because a patentee may define his own terms, give a claim term a different meaning than the term would otherwise possess, or disclaim or disavow the claim scope. *Phillips*, 415 F.3d at 1316. In these situations, the inventor's lexicography governs. *Id*. The specification may also resolve ambiguous claim terms "where the ordinary and accustomed meaning of the words used in the claims lack sufficient clarity to permit the scope of the claim to be ascertained from the words alone." *Teleflex, Inc.*, 299 F.3d at 1325. But, "'[a]lthough the specification may aid the court in interpreting the meaning of disputed claim language, particular embodiments and examples appearing in the specification will not generally be read into the claims.'" *Comark Commc'ns, Inc. v. Harris Corp.*, 156 F.3d

1182, 1187 (Fed. Cir. 1998) (quoting *Constant v. Advanced Micro-Devices, Inc.*, 848 F.2d 1560, 1571 (Fed. Cir. 1988)); *see also Phillips*, 415 F.3d at 1323. The prosecution history is another tool to supply the proper context for claim construction because a patent applicant may also define a term in prosecuting the patent. *Home Diagnostics, Inc., v. Lifescan, Inc.*, 381 F.3d 1352, 1356 (Fed. Cir. 2004) ("As in the case of the specification, a patent applicant may define a term in prosecuting a patent.").

Although extrinsic evidence can be useful, it is "'less significant than the intrinsic record in determining the legally operative meaning of claim language.'" *Phillips*, 415 F.3d at 1317 (quoting *C.R. Bard, Inc.*, 388 F.3d at 862). Technical dictionaries and treatises may help a court understand the underlying technology and the manner in which one skilled in the art might use claim terms, but technical dictionaries and treatises may provide definitions that are too broad or may not be indicative of how the term is used in the patent. *Id.* at 1318. Similarly, expert testimony may aid a court in understanding the underlying technology and determining the particular meaning of a term in the pertinent field, but an expert's conclusory, unsupported assertions as to a term's definition are entirely unhelpful to a court. *Id.* Generally, extrinsic evidence is "less reliable than the patent and its prosecution history in determining how to read claim terms." *Id.*

DISCUSSION

1. Construction of "General Purpose Programmable Media Processor"

Term	MicroUnity's Proposal	Defendants' Proposal
"general purpose programmable media processor" '840 Pat., claim 1; '061 Pat., claim 1, 31; '318 Pat., claim 1, 2, 5, 8.	"a processor having an execution unit capable of operating on different media types and data sizes using a single general purpose instruction set"	"one or more integrated circuit chips that process media data, having a single general-purpose processor but not specialized media processing hardware, and using a single general purpose instruction set"

The Court construes the term **"general purpose programmable media processor"** (GPPMP) to mean **"a single processor having an execution unit capable of operating on different media types and data sizes using a single general purpose instruction set."**

The parties agree as to "using a single general purpose instruction set." As to the term "single processor," Defendants requested single to be included in the portion of their construction which states "a single general-purpose processor." At the hearing, MicroUnity agreed to use "single" within its construction. Thus, the parties agree also as to the concept of "single."

Though the parties do not debate that the processor can be "one or more integrated circuit chips" and the specification describes multi-chip embodiments (for example, Figure 19 of the '840 patent), the inclusion of such language does not assist in defining a GPPMP, and appears to have the potential to create confusion for the jury. Thus, the Court does not find such language beneficial for use in the construction.

The primary dispute between the parties relates to whether the processor can include "specialized media processing hardware." Defendants request a construction that excludes "specialized media processing hardware."

Though the Defendants assert that input/output circuitry is not part of the processor, the specification clearly describes embodiments that include the I/O as part of the processor: "Alternatively, Fig. 6 shows a presently preferred integrated general purpose media processor 12. The integrated processor includes on-board memory and I/O 86." '840 Pat. at 11:42-44 and Fig. 6. As shown in Figure 7, I/O circuitry may be Nyquist Sampled I/O 128. *Id*. at Figure 7 and 18:17-36. The specification describes this I/O as a digital sampling I/O, thus circuitry that processes the data to some extent. Though the specification illustrates and describes different

types of data (audio, video, network, etc.) being sampled by the Nyquist Sampled I/O block 128, the specification does not state that the I/O block cannot include specialized processing hardware. *Id*. at 18:17-37 and Fig. 7. Elsewhere, in fact, the specification appears to imply that specialized circuitry may exist. With regard to the embodiment of Figure 19, the I/O block 216 is illustrated with separate circuit blocks for audio, video and network. *Id.* at Fig. 19, 26:64-27:7. It is noted that all the circuit blocks of Figure 19 may implemented in a single integrated processor. *Id.* at 27:15-24. Though the specification does not provide detail as to these circuit blocks, the teaching as a whole cannot be said to exclude from the processor hardware circuitry that performs specialized media processing. Such a limitation would conflict with what appears to be an I/O that does have circuitry blocks that are specialized for processing different media and data types.

Defendants assert that the specification teaches away from the use of specialized media processors. In particular, Defendants point to Figure 5 and the associated text. Figure 5 describes a system in which separate media processors 76, 78, 80 and 82 are utilized either as standalone integrated circuits or combined in a single chip ("the future of the old way"). Defendants assert that this is contrasted with the invention described with reference to Figure 6 in which a single processor 12 is provided for processing multiple types of media data. Defendants assert that such disparagement in the specification amounts to a disavowal such that a proper construction requires excluding "specialized media processing hardware." However, as noted above, Defendants' assertions would conflict with teachings in the specification in which a processor may include specialized media circuitry for processing different media types in the I/O circuitry. Furthermore, the prior art that Defendants point to illustrates an approach in which no single media processor includes shared functionality of different types of media. Yet Defendants

construction would exclude either approach of Figure 6 ("the unified way" or "the future of the unified way") in which audio, video, graphics and network information are processed in a single processor, merely if some additional specialized processor circuitry is added to the system. Such an interpretation neither conforms with the law regarding the proper construction of claims using the open transition "comprising," nor with a natural reading of the specification as a whole. As noted in the reexamination proceedings of the '840 and '061 patents, rather than emphasizing that every function must be performed by a general purpose media processor, emphasis was given to allowing flexibility to perform not all but at least some of a variety of media functions:

> The ability to dynamically partition integer and floating point data permits more efficient processor operation, and moreover, allows the designer to make strategic decisions about which operations are to be performed by the processor and which are to be relegated to peripheral processors or ASICS.

('840 RE Resp. at 4, Dkt No. 433-32.)

> To overcome problem the inventors sought a solution in which a general purpose processor was capable of performing at least certain aspects of media processing on a plurality of different types of media data. 2:49-3:40. As an example, the media processor could be configured to perform those aspects of media processing that are the most susceptible to modification, or which the media processor is best-suited to perform for a particular application.

('061 RE Resp. at 13, Dkt. No. 433-31.)

> The ability to dynamically partition the presentation, transmission and storage information in the media processor permits a more effective use of the media processor, and moreover, allows the designer to make strategic decisions about which operations are to be performed by the media processor and which are to be relegated to conventional ASIC devices.

('061 RE Resp. at 14, Dkt. No. 433-31.)

Taken as a whole, the intrinsic record emphasizes combining processor functions, but does not mandate that every specialized processing circuit must always be eliminated. Thus,

emphasis is given to an execution unit that can operate on different media types and sizes. '840 Pat. at 3:65-4:41, 5:23-38, 11:19-26, 11:50-12:55.

2. Construction of "Multi-Precision Execution Unit"

Term	MicroUnity's Proposal	Defendants' Proposal
"multi-precision execution unit" '840 Pat., claim 1; '061 Pat., claims 1, 15, 31, 34, 44, 47; '765 Pat., claims 1, 2, 7, 9, 11, 12, 14, 15, 16, 17, 24, 26; '217 Pat., claims 1, 2, 7, 9, 11, 12, 14, 16, 17, 24, 26, 32.	"a unit that receives instructions and executes the instructions to perform simultaneous parallel operations on the plurality of media data streams, each of a width up to the width of the data path"	"the core of a processor that includes a multi-precision arithmetic unit, a programmable switch, and an extended mathematical element"

The Court finds that the term **"multi-precision execution unit"** means "**a unit that receives instructions and executes the instructions to perform simultaneous parallel operations on the plurality of media data streams, each of a width up to the width of the data path."**

The first issue presented by the parties is whether the "multi-precision execution unit" is the "core" of a processor. Defendants argue that the specification and the reexamination history require that the multi-precision execution unit be "the core of a processor." (Resp. Br. 12-13, Dkt. No. 467.) At the hearing, Defendants explained that the term "core" itself does not have any particular technical meaning, but should be included in the term's construction because it emphasizes the centrality of the execution unit to the processor, and makes "clear that the multi-precision arithmetic unit, programmable switch, and extended mathematical element exists in one unit [that is] the center of the processor." (Tr. at 63:22-65:16, Dkt. No. 488.)

MicroUnity responds that the only reference to the execution unit being the "core" of the processor is in the description of the preferred embodiment, and that it argued during reexamination that the execution unit was the "core" of a processor while explaining to the

examiner that the execution unit was a part of the processor, and not the entire processor. (Reply Br. 4-5, Dkt. No. 474; Tr. at 43:4-44:14.) MicroUnity also argues that the phrase "core of a processor" should not be added to the term's construction because there is no agreed meaning for the term "core," and that the term "core" itself is likely to cause juror confusion. (Tr. at 44:15-45-7.)

The Court rejects including the phrase "core of a processor" in the construction of "multi-precision execution unit," because (1) it is not clearly supported by the specification, (2) it is not compelled by the patentee's statements made during the reexamination of the '840 and '061 patents, (3) and the term "core" itself is ambiguous and likely to cause confusion. The specification does include the statement that"[t]he core of the integrated general purpose media processor 12 comprises execution unit 100." '840 Pat. at 11:52-54. However, that statement is made in the context of describing the preferred embodiment depicted in Figure 7. *See* '840 Pat. at 11:51-52 (The preceding sentence reads: "One presently preferred embodiment of an integrated general purpose media processor 12 is shown in FIG. 7.").

The prosecution history shows that the patentee used the term "core" during prosecution to argue that the execution unit is a part of the processor, and to argue that a component outside of a processor could not be part of the execution unit. During the reexamination of '840 patent, the examiner rejected claims directed to the preferred embodiment over the MVP reference. The patentee held the view that the examiner's broadest reasonable construction of the term "multi-precision execution unit" improperly equated the execution unit with the general purpose media processor as a whole. (Dkt. No. 467-9 at 9-10.) The patentee pointed to the "core" discussion in the specification while making the argument that "interpreting the execution unit to be the entire media processor 12 is clearly inconsistent with the Specification." (*Id.* at 12.) During the

reexamination of the ‘061 patent, the patentee relied on the same discussion of the execution unit being part of the processor’s “core” to show that a cross-bar switching network between processors, which was described in a prior art reference, could not be considered a part of the claimed execution unit. (Dkt. No. 467-10 at 31-32.) In the full context of the arguments that were made, all that the prosecution history shows is that the term “core” was used to show the relationship between the execution unit and the general purpose media processor, which is that the execution unit is a part of the general purpose media processor.

As neither party is able to give a clear definition of what the term “core” means, or can explain how the jury should properly apply the term if included in the construction, the Court agrees that including “core” in the construction would introduce ambiguity and juror confusion. For all these reasons, the Court declines to add the “core” limitation to the construction of “multi-precision execution unit.”

The next issue is whether the “multi-precision execution unit” contains an arithmetic unit, a switch, and an extended mathematical element. The Court finds that the term “multi-precision execution unit” does not include the additional elements consisting of “a multi-precision arithmetic unit, a programmable switch, and an extended mathematical unit.” Starting with the claim language, MicroUnity brings to the Court’s attention that some claims specifically recite the additional elements while others do not. For example, claim 1 of the ‘061 patent recites “a multi-precision execution unit.” Claim 15, which depends upon claim 1, further recites “wherein the multi-precision execution unit comprises a dynamically partitionable arithmetic unit, a register controllable cross-bar switch, and an extended mathematical element.” Under the doctrine of claim differentiation, a rebuttable presumption arises against a claim construction that gives the same meaning or claim scope to separate claims that use different words or phrases.

See Seachange Int'l, Inc. v. C-COR, Inc., 413 F.3d 1361, 1368-69 (Fed. Cir. 2005). The presumption is strongest "'where the limitation sought to be "read into" an independent claim already appears in a dependent claim.'" *Id.* (quoting *Liebel-Flarsheim Co. v. Medrad, Inc.*, 358 F.3d 898, 910 (Fed. Cir. 2004)).

In this instance, the Court finds that the evidence does not rebut the presumption. Although there is a description of an embodiment "[a]ccording to the apparatus of the invention" where the additional elements are described, other portions of the specification describing "another aspect of the invention" or "yet another aspect of the invention" do not similarly recite the additional elements. *See* '840 Pat. at 4:10-24, 4:25-41, and 5:23-38.

The reexamination history also supports not construing the term to include the additional elements. In claims where the additional elements were expressly recited in the claims, MicroUnity distinguished prior art on the basis that the additional elements were not present in the prior art. (*See* '061 RE at 27, Dkt. No. 467-10 (discussing claims 15-17, 35, 37-39, 44 and 47).) However, where the claim did not recite the additional elements, MicroUnity distinguished the prior art on other grounds and did not argue that the additional elements were inherently required to be present in the prior art by the use of the term "multi-precision execution unit." (*See Id.* at 37-40 (discussing claims 31, 32, and 40-42).) The Court finds that it is not proper to read in the additional elements into the claim term.

3. Construction of "Multi-Precision Arithmetic Unit"

Term	MicroUnity's Proposal	Defendants' Proposal
"multi-precision arithmetic unit" '840 Pat., claim 1, 11; '318 Pat., claim 8.	"a unit that can perform addition, subtraction, multiplication, division, and other integer and floating point arithmetic operations on data streams of varying sizes"	"a logic unit that can perform group operations on data streams of varying sizes for each of (1) addition; (2) subtraction; (3) multiplication; (4) division; (5) an integer operation; and (6) a floating point operation"

The Court finds that the term **"multi-precision arithmetic unit"** means **"a unit that can perform group operations of addition, subtraction, multiplication, division, and other integer and floating point arithmetic operations on data streams of various sizes."**

MicroUnity's proposed construction is the same construction adopted by the *Intel* court. (*Intel* Claim Construction Order at 12, *MicroUnity Sys. Eng'g, Inc. v. Intel Corp., et al.*, Case No. 2:04-CV-120, Dkt. No. 167.) Defendants' proposed construction is very similar. The parties agree that a "multi-precision arithmetic unit" must perform all six operations listed in the proposed constructions. The parties also agree that the operations are "group operations." (Tr. at 71:2-16.) The only dispute is whether the arithmetic unit must be operable to perform each type of group operation at multiple precisions.[1]

MicroUnity argues that the claims and the specification do not support requiring the "multi-precision arithmetic unit" to be capable of performing group operations at different precisions for each of the arithmetic operations. (Opening Br. 16-18, Dkt. No. 443.) According to MicroUnity, claim 36 of the '061 patent shows that "the ability to dynamically partition floating-point (or integer) data of one precision is sufficient, as long as the precision of the integer operation and floating-point operation are different." Claims 15, 35, and 36 of the '061 patent recite:

> 15. The media processor defined in claim 1, wherein the multi-precision execution unit comprises a dynamically partitionable arithmetic unit, a register controllable cross-bar switch, and an extended mathematical element.
>
> * * *

[1] At the hearing, it became apparent that there is no real dispute over whether the "multi-precision arithmetic unit" and "dynamically partitionable arithmetic unit" terms should refer to a "unit" or a "logic unit." Both parties agree that the terms are not limited to a single circuit. These issues were initially disputed because each side mistakenly thought that the opposing side's proposal would limit the terms to single circuits. (*See* Tr. 99:16-100:7.)

> 35. The system defined to claim 15, wherein said dynamically partitionable arithmetic unit, in response to a single programmed instruction, dynamically partitions integer or floating point operands stored in the register file into partitioned data fields, and performs an arithmetic operation on each of the partitioned data fields to produce a catenated result.
>
> 36. The system defined in claim 35, wherein said dynamically partitionable arithmetic unit dynamically partitions integer data of a first precision and performs an arithmetic operation on each of the partitioned integer data fields, and dynamically partitions floating point data of a second precision different than the first precision and performs an arithmetic operation on each of the partitioned floating point data fields.

MicroUnity contends that claim 36 would be meaningless if both the integer data and the floating-point data have to be partitioned in at least two different precisions because there would always be at least one partitioned floating-point datum at a different precision from one of the partitioned integer data. (Opening Br. 17-18.) MicroUnity also argues that Defendants' construction, which requires the ability to perform group floating-point operations at two precisions, is inconsistent with the prosecution history where MicroUnity "noted that the 'multi-precision arithmetic unit' required 'the ability to perform at least one group floating point operation.'" (*Id.* at 18.)

Defendants respond that MicroUnity's arguments should be disregarded because claim 36 was added during reexamination, and the statement regarding "at least one" was made after the *Intel* construction issued. (Resp. Br. 20.) Defendants argue that it would be improper to permit a dependent claim added during reexamination to broaden the scope of the pre-existing claims. (*Id.*) The problem with Defendants' argument is that it assumes the examiner failed to appreciate the allegedly broadening nature of claim 36, and then erred by allowing the claim to issue. It is also possible that the examiner interpreted the claims in the same manner as MicroUnity, and found that claim 36 was narrower in scope than the existing claims. The only support

Defendants' cite for their claim construction is the "plain meaning" of the *Intel* construction, and a portion of the specification that the Court does not agree clearly supports limiting the claims as Defendants suggest. (*See* Resp. Br. 19-20.)

4. Construction of "Dynamically Partitionable Arithmetic Unit"

Term	MicroUnity's Proposal	Defendants' Proposal
"dynamically partitionable arithmetic unit" '061 Pat., claim 15, 35, 36, 47.	"the arithmetic unit can be divided into a variable number of elements"	"a multi-precision arithmetic unit (as construed herein) that is capable of being divided widthwise into a variable number of logic units"

The Court finds that the term **"dynamically partitionable arithmetic unit"** means **"an arithmetic unit that can partition data into fields of variable width."**

The parties dispute whether it is the data or the arithmetic unit itself that must be partitionable. Defendants contend that "[t]he plain language of the claims require the arithmetic unit, not the data, to be partitionable." (Resp. Br. 21-22.) Defendants cite claim 15 of the '061 patent as the only example:

> 15. The media processor defined in claim 1, wherein the multi-precision execution unit comprises a dynamically partitionable arithmetic unit, a register controllable cross-bar switch, and an extended mathematical element.

Defendants also rely on the following portion of the specification:

> Many of the logic blocks themselves can also [be] replaced with a single multi-precision arithmetic unit, *which can be internally partitioned under software control* to perform addition, multiplication, division, and other integer and floating point arithmetic operations on symbol streams of varying widths, while sustaining the full data throughput of the memory hierarchy.

'840 Pat. at 2:58-65 (emphasis added).

MicroUnity argues that dependent claims 35, 36, and 37 "make clear that the data is partitioned, and not the arithmetic unit" because "[t]hese claims all require partitioning data into

fields for further operation" and that "[t]here is nothing in these claims that dictates dividing the arithmetic unit into logic units." (Opening Br. 19-20.)

The specification's teaching that arithmetic units "can be internally partitioned under software control" does not clearly speak to whether it is the data or the arithmetic unit that is partitioned. On balance, the Court is persuaded that it is the data that must be partitioned, and that the arithmetic unit is not physically partitioned. While the specification clearly discusses partitioning the data, it does not clearly discuss partitioning the arithmetic unit hardware physically.

The parties dispute whether a "dynamically partitionable arithmetic unit" must include a "multi-precision arithmetic unit." The term itself only appears in some claims of the '061 patent. While the term "dynamically partitionable arithmetic unit" is not used in the specification, Defendants contend that several portions of the specification show that "dynamically partitionable" is linked to a "multi-precision execution unit":

> The execution unit includes a dynamically partitionable multi-precision arithmetic unit, programmable switch and programmable extended mathematical element.

'840 Pat. at Abstract.

> Many of the logic blocks themselves can also [be] replaced with a single multi-precision arithmetic unit, which can be internally partitioned under software control to perform addition, multiplication, division, and other integer and floating point arithmetic operations on symbol streams of varying widths, while sustaining the full data throughput of the memory hierarchy.

'840 Pat. at 2:58-65.

> According to the apparatus of the invention, an execution unit is provided that maintains substantially peak data throughput in the unified execution of multiple media data streams. The execution unit includes a data path, and a multi-precision arithmetic unit coupled to the data path and capable of dynamic partitioning based on the elemental width of data received from the data path.

‘840 Pat. at 4:10-16.

Defendants also argue that MicroUnity disclaimed any “dynamically partitionable arithmetic unit” that does not have a “multi-precision arithmetic unit” because it applied “the definition of ‘multi-precision arithmetic unit’—which, as discussed above, requires six specific operations—to the ‘dynamically partitionable arithmetic unit’ term.” (Resp. Br. 20-21.)

First, Defendants rely on MicroUnity’s response to an office action from the reexamination of the ‘061 patent. (‘061 RE 02/22/07 Resp., Dkt. No. 467-4.) Defendants argue that MicroUnity distinguished “TI prior art because it did not disclose group multiplication or division operations.” (Resp. Br. 21.) However, the cited portion merely provides a background discussion of the TI prior art, and does not tie the distinguishing aspects to the claim term at issue. (*Id.* at 28-32.) Where there is a discussion of claim 15 of the ‘061 patent, MicroUnity noted that the TI prior art does not disclose the register controllable cross-bar switch and the extended mathematical element. (*Id.* at 37.)

Second, Defendants rely on a summary of an examiner interview from the reexaminations of the ‘840 and ‘061 patents. (Interview Summary, Dkt No. 467-11.) Defendants argue that MicroUnity distinguished “Motorola prior art because no ‘group multiply or divide’ operations” were present.” (Resp. Br. 21-22 (citing Interview Summary at 10).) While the cited portion would support finding that a “dynamically partitionable arithmetic unit” must be capable of performing group multiply or divide operations, this does not imply that the other limitations of a “multi-precision arithmetic unit” necessarily apply to a “dynamically partitionable arithmetic unit.”

Defendants further argue that MicroUnity distinguished “the ‘features of a multi-precision arithmetic unit as set forth in claim’ from prior art because the prior art lacked group

multiply or divide operations." (*Id.* at 21.) The Court finds that this portion of the interview summary does not support a disclaimer because it is not sufficiently clear and unambiguous. As MicroUnity points out, there is no "multi-precision arithmetic unit" recited in claim 15 of the '061 patent, but the term appears in the '840 claims also discussed at the same interview. (Tr. 85:20-87:7.) The heading of the section where the statement appears reads "A Dynamically Partitionable Arithmetic Unit." (Interview Summary at 14.) Thus, it appears to the Court that "multi-precision arithmetic unit" appeared by mistake, and should have recited "dynamically partitionable arithmetic unit." This is consistent with the remainder of the interview summary because the same point of distinction (group multiplication and division) was made a few pages earlier for the "dynamically partitionable arithmetic unit" term as applied to the Motorola prior art. (*Id*. at 10.)

5. Construction of the 128-Bit Data Path Terms

Term	MicroUnity's Proposal	Defendants' Proposal
"the width of the data path is at least 128 bits" "the data path is at least 128 bits wide" '061 Pat., claim 14; '318 Pat., claim 5.	"at least 128-bit wide buses and circuit elements that convey data"	"the data path has width of 128 bits or more throughout"

The Court finds that the 128-bit data path terms mean **"the claimed data path has a width of 128 bits or more."**

The parties agree that a "data path" means "buses and circuit elements that convey data." (Tr. 106:15-21.) At the hearing, the parties appeared to have a shared understanding of what it means for a data path to be "at least 128 bits," but were unable to succinctly describe the meaning. Defendants want it made clear that a data path must be "at least 128 bits" along its

entire length. (Tr. 106:22-107:6.) This explains Defendants' suggestion of including "throughout" in their proposed construction. Plaintiff is concerned by the use of "throughout" because there may be more than one data path between any two given components, and to satisfy the open-ended transition "comprising," it is only necessary that one of the data paths be 128 bits wide. (Tr. 104:10-105:19 and 109:11-23.) In other words, if an accused device has a 128-bit wide bus and a 64-bit wide bus between two components, the claim is satisfied by the 128-bit wide bus. The parties agree that each other's propositions are correct.

In an attempt to capture this understanding, the Court construes the terms to mean "the claimed data path has a width of 128 bits or more." In other words, there must be at least one data path between the two elements described by the relevant surrounding claim language that has a width of 128 bits or more along the length of the data path.

6. Construction of "Elemental Width"

Term	MicroUnity's Proposal	Defendants' Proposal
"elemental width" '217 Pat., claim 32, 35.	"an instruction that specifies the elemental width of the floating point data and a floating point arithmetic operation"	"instruction that selects the width of the floating-point data from several available widths for an identified group floating-point arithmetic operation"

The Court finds that **"elemental width"** means **"an instruction that specifies an elemental width of the floating point data and a floating point arithmetic operation."**

MicroUnity contends that the claim language only requires specifying a width, and does not require selecting "from several available [floating point] widths." (Opening Br. 21-22.) MicroUnity points to several dependent claims that explicitly require different floating-point widths, and therefore there is a rebuttable presumption against Defendants' construction arising from the doctrine of claim differentiation. (*Id.*) As one example, claim 37 of the '217 patent

recites that "the execution unit is capable of performing group floating-point arithmetic operations on floating-point data of at least two different elemental widths." (*Id.*)

Defendants argue that claim 35 of the '217 patent (on which claim 37 depends) was distinguished over the prior art on the grounds that "claim 35 partitions data into 'a variable number of data elements corresponding to partitioned fields' that are 'used in floating-point operations.'" (Resp. Br. 26.)

The Court has reviewed the statements made by MicroUnity during the reexamination of the '217 patent, and finds that there was no clear and unambiguous disclaimer that justifies Defendants' proposed limitation of "[selecting] the width of the floating-point data from several available widths." In discussing the application of the Argawal reference to claim 1 of the '217 patent, MicroUnity argued that "Agarwal does not teach, disclose, or suggest that data be partitioned into a variable number of data elements no wider than the data path based on an elemental width corresponding to the size of the data elements." ('217 RE 08/25/2003 at 20, Dkt. No 467-13.) MicroUnity then extended the argument by stating: "[f]urthermore, Agarwal does not teach performing group floating-point operations on multiple operands that are derived from dynamically partitioned data." (*Id.*) In response to the rejection of claim 35, MicroUnity relied on these arguments: "[a]s discussed above, Agarwal does not teach, disclose, or suggest that data in an operand register be partitioned into a variable number of data elements based on a dynamic elemental width specified in the same instruction for floating-point arithmetic operation." (*Id.* at 22.) In the context of the response, the Court finds that MicroUnity's statements focused on the absence of partitioning in the Agarwal reference, and do not support requiring "[selecting] the width of the floating-point data from several available widths."

7. Construction of "Group Floating-Point Operation"

Term	MicroUnity's Proposal	Defendants' Proposal
"group floating point operation[s]" '765 Pat., claims 1, 7, 9; '356 Pat., claims 18, 59, 60, 74; '217 Pat., claims 1, 7, 9, 42; '287 Pat., claim 12, 21; '973 Pat., claim 1, 12; '806 Pat., claim 1, 5, 10, 12.	"floating point operations simultaneously applied to a group of partitioned operands"	"SIMD floating point operations simultaneously applied to a group of partitioned operands"

The Court finds that **"group-floating point operation[s]"** means **"floating point operations simultaneously applied to a group of partitioned operands."**

MicroUnity proposes the *Intel* court's construction for this term, with the addition of "simultaneously" as proposed by the Defendants. (Opening Br. 22.) Defendants request that "SIMD," which refers to "single instruction multiple data," be included in the construction because the specification uses "SIMD" and "group" interchangeably. (Resp. Br. 27.) At the hearing, Defendants also suggested that including "SIMD" was necessary to prevent MicroUnity from arguing that so-called "MIMD" (multiple instruction multiple data) operations are group operations. (Tr. 118:7-119-20.)

MicroUnity is opposed to including "SIMD" because the technical term has multiple meanings in the art and one of those meanings "would impose a multiprocessor requirement on claims that otherwise read on a single processor." (Opening Br. 22-23 (citing technical definitions for "SIMD" operations).) MicroUnity believes that the term would confuse jurors, and that "SIMD" itself would require further construction. (Reply Br. 9.) MicroUnity has stated that it does not contend that this term covers "MIMD" operations. (Tr. 119:23-120:13.)

The Court finds that the term "SIMD" should not be included in the construction. The term appears to be susceptible to multiple meanings, which may lead to juror confusion if no

further construction is provided. Given that the term was used "interchangeably" with group, the Court does not believe it adds much to the understanding of the claim language.

8. Construction of "Partitioning a . . . Register"

Term	MicroUnity's Proposal	Defendants' Proposal
"partitioning a . . . register" '366 Pat., claim 1.	"separating the data in a register into distinct fields"	"dividing a register widthwise into a variable number of separately accessible read/write fields"

The Court finds that **"partitioning a . . . register"** means **"dividing a register width-wise,"** which is the portion of the parties' proposed constructions that is agreed to.[2]

MicroUnity opposes the additional limitation of dividing a register "into a variable number of separately accessible read/write fields" because there is no support for the limitation in the claims or the specification, and it would not cover the preferred embodiment. (Opening Br. 24.) MicroUnity argues that the additional limitation is inconsistent with the preferred embodiment because the patent discloses that "no portion of a register smaller than 64 bits can be separately read or written," and yet also discloses that "the data in registers [can] be partitioned into 8-bit, 16-bit, or 32-bit data elements." (*Id.* (citations omitted).) MicroUnity also notes that "variable" partitioning is not a requirement of every claim where the "partitioning" term appears, which is another reason to not adopt Defendants' construction. (Tr. 122:9-16 and 123:6-11.)

Defendants argue that the portion of the appendix cited by MicroUnity refers to partitioning data and not partitioning registers. (Resp. Br. 24.) Defendants argue that Figure 12B of the '765 patent "illustrates how registers ('rd' and 'rb') can be partitioned widthwise into a variable number of read/write fields," and that it "discloses computer code for a wide-switch instruction that partitions 'full-width' 128-bit registers into separately accessible 128-bit or 64-

[2] At the hearing, MicroUnity dropped its original proposed construction, and adopted Defendants' proposed construction in part: "dividing a register width-wise." (Tr. 120:18-23.)

bit read/write fields . . . ." (Resp. Br. 23-24 (citations omitted).) While Defendants' argument appears to accurately summarize the cited portions of the specification, it does not support a requirement that a register be divided into "*a variable number* of separately accessible read/write fields." The Court rejects Defendants additional limitation because Defendants have not adduced sufficient evidence to support it.

9. Construction of "Register[s]"

Term	MicroUnity's Proposal	Defendants' Proposal
"register[s]" '061 Pat., claims 1, 15, 17, 31, 35, 37, 38, 44, 47; '765 Pat., claims 1-3, 7, 11, 12, 14, 15, 17, 20-24; '356 Pat., claims 18, 36, 44, 59-61, 68; '366 Pat., claims 1, 5, 6, 9, 13, 14; '217 Pat., claims 1, 2, 7, 11, 12, 14, 17, 24, 32, 35, 42, 46; '367 Pat., claims 1, 32, 42, 47.	"hardware storage locations that are directly addressable"	"the smallest portion of a register file that is separately readable by a user/programmer"

The Court finds that the terms **"register"** and **"registers"** should be given their **plain and ordinary meaning**. The Court rejects Defendants contention that reading "a register" means reading one and only one register.

There does not appear to be any genuine dispute that the terms "register" and "registers" have a meaning that is well known to persons of ordinary skill in the art. (*See* Tr. 133:9-19 and 134:5-6.) Instead, the issue Defendants raise is whether or not "register" covers register pairs. (Tr. 135:23-136:3.) At the hearing, Defendants explained that there is "a concept [described] in the patent of register pairs where you are actually reading from two different registers or storing in two different registers" and "[y]ou can address a register pair by addressing one register and saying just keep reading [to read the contents from both]. [Defendants] want to make it clear that

the register is the register itself and not what else you might want to read with it." (Tr. 133-24-134:1 and 134:11-15.)

MicroUnity argues that "[t]he claims, the specification, everything makes it very clear that when you read, you can read a 64-bit quantity, you can read a 128-bit quantity. If you have physical registers that are 64 bits wide and you do a read register operation on 128 bits, you are reading a register pair." (Tr. 135:8-13 (citing the '366 patent).) MicroUnity also argues that Defendants' construction would exclude the preferred embodiment because, in the preferred embodiment, registers may be read as both 64 bits and 128 bits. (Opening Br. 25.)

The Court agrees that the terms "register" and "registers" do not exclude performing an operation using register pairs. The appendix describes a processor that has 64-bit registers, and discloses that a 128-bit register read or write operation is performed by reading or writing a pair of 64-bit registers:

```
Definition
def val ← RegRead(rn, size)
    case size of
        64:
            val ← REG[rn]
        128:
            if rn0 then
                raise ReservedInstruction
            endif
            val ← REG[rn+1] || REG[rn]
    endcase
enddef
```

('840 Pat. Appx. at 24.) There is no evidence that MicroUnity disclaimed or disparaged the technique of using register pairs to provide the same functionality of a larger physical register by using register pairs. Moreover, the Court is persuaded that Defendants' construction would exclude the preferred embodiment from the claims.

10. Construction of "Subfield"

Term	MicroUnity's Proposal	Defendants' Proposal
"subfield" '367 Pat., claims 1, 32, 42, 47; '973 Pat., claims 16-20.	"a number of contiguous bits of a field"	"a contiguous collection of bits having a width less than the elemental width of the data element"

The Court finds that **"subfield"** means **"a number of contiguous bits forming all or part of a field."**

Defendants assert that the plain meaning of "subfield" is less than a full field, and that the patents repeatedly refer to a subfield in the context of a collection of bits less than a full field: "a sequential field of bits 270 can be divided into constituent sub-fields" ('840 Pat. at 12:19-22); "separate sub-fields 272a-272d from a larger bit field 274 can be combined to form a contiguous or sequential field of bits 270" (*Id.* at 12:25-27); and "Figure 8B of the '840 patent illustrates that 2:10-cv-00091 'sub-fields 272a-d' are less than the width of the four partitioned sub-components of 'larger bit field 274'" (Resp. Br. 29 (citing '840 Pat.)).

MicroUnity argues that the plain meaning of "subset" is "a set each element of which is an element of a specified other set." (Reply 10 (citing *IBM Dictionary of Computing*, at 660 (10th ed. 1993).) In other words, under the plain and ordinary meaning of "subset," one subset of a given set is the set itself. Therefore, a "subfield" may be the entire field, which would be improperly excluded under Defendants' construction.

MicroUnity also argues that "[t]he '840 Appendix shows that the 'subfield,' or contiguous bits within a field that are shifted, can be the same size as the field." (Opening Br. 25-26.) This is because the appendix shows that a group shift left immediate instruction "will cause the entire field to be shifted when the shift amount is zero (an operation useful for depositing a copy of the entire register into a different destination register)." (*Id.*) Defendants'

response to MicroUnity's shift argument ("When the shift amount is zero . . . the data is not shifted at all but is merely moved from one register to another." (Resp. Br. 30)) is foreclosed by the language of claim 1 of the '367 patent ("the shift amount configurable to an amount inclusively between zero and one less than the elemental width").

The Court finds that "subfield" means "a number of contiguous bits forming all or part of a field." The Court is persuaded that this construction is consistent with the plain and ordinary meaning of "subfield" and that the zero-shift embodiment described in the specification and claimed by the '367 patent should be covered by the construction.

CONCLUSION

The Court adopts the above constructions. The parties are ordered that they may not refer, directly or indirectly, to each other's claim construction positions in the presence of the jury. Likewise, the parties are ordered to refrain from mentioning any portion of this opinion, other than the actual definitions adopted by the Court, in the presence of the jury. Any reference to claim construction proceedings is limited to informing the jury of the definitions adopted by the Court.

SIGNED this 7th day of March, 2013.

ROY S. PAYNE
UNITED STATES MAGISTRATE JUDGE